Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

J. Peterson
NeuStar
C. Jennings
Cisco
E. Rescorla
RTFM, Inc.

October 21, 2013

Authenticated Identity Management in the Session Initiation Protocol (SIP)
draft-jennings-stir-rfc4474bis-00

Abstract

The baseline security mechanisms in the Session Initiation Protocol (SIP) are inadequate for cryptographically assuring the identity of the end users that originate SIP requests, especially in an interdomain context. This document defines a mechanism for securely identifying originators of SIP requests. It does so by defining new SIP header fields for conveying a signature used for validating the identity, and for conveying a reference to the credentials of the signer.

Table of Contents

1. Introduction

This document provides enhancements to the existing mechanisms for authenticated identity management in the Session Initiation Protocol (SIP, RFC 3261 [RFC3261]). An identity, for the purposes of this document, is defined as either a SIP URI, commonly a canonical address-of-record (AoR) employed to reach a user (such as 'sip:alice@atlanta.example.com'), or a telephone number, which can be represented as either a TEL URI or as the user portion of a SIP URI.

RFC 3261 [RFC3261] stipulates several places within a SIP request where a user can express an identity for themselves, notably the user-populated From header field. However, the recipient of a SIP request has no way to verify that the From header field has been populated appropriately, in the absence of some sort of cryptographic authentication mechanism.

RFC 3261 [RFC3261] specifies a number of security mechanisms that can be employed by SIP user agents (UAs), including Digest, Transport Layer Security (TLS), and S/MIME (implementations may support other security schemes as well). However, few SIP user agents today support the end-user certificates necessary to authenticate themselves (via S/MIME, for example), and furthermore Digest authentication is limited by the fact that the originator and destination must share a prearranged secret. It is desirable for SIP user agents to be able to send requests to destinations with which they have no previous association -- just as in the telephone network today, one can receive a call from someone with whom one has no

previous association, and still have a reasonable assurance that the person's displayed calling party number (and/or Caller-ID) is accurate. A cryptographic approach, like the one described in this document, can provide a much stronger and less spoofable assurance of identity than the telephone network provides today.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [RFC2119] and RFC 6919 [RFC6919].

3. Background

The usage of many SIP applications and services is governed by authorization policies. These policies may be automated, or they may be applied manually by humans. An example of the latter would be an Internet telephone application that displays the calling party number (and/or Caller-ID) of a caller, which a human may review (making a policy decision) before answering a call. An example of the former would be a voicemail service that compares the identity of the caller to a whitelist before determining whether it should allow the caller access to recorded messages. In both of these cases, attackers might attempt to circumvent these authorization policies through impersonation. Since the primary identifier of the sender of a SIP request, the From header field, can be populated arbitrarily by the controller of a user agent, impersonation is very simple today. The mechanism described in this document provides a strong identity system for SIP requests in which authorization policies cannot be circumvented by impersonation.

This document proposes an authentication architecture for SIP in which requests are processed by a logical authentication service that may be implemented as part of a user agent or as a proxy server. Once a message has been authenticated, the service then adds new cryptographic information to requests to communicate to other SIP entities that the sending user has been authenticated and its use of the From header field has been authorized.

But authorized by whom? Identities are issued to users by authorities. When a new user becomes associated with example.com, the administrator of the SIP service for that domain will issue them an identity in that namespace, such as alice@example.com. Alice may then send REGISTER requests to example.com that make her user agents eligible to receive requests for sip:alice@example.com. In some cases, Alice may be the owner of the domain herself, and may issue herself identities as she chooses. But ultimately, it is the

controller of the SIP service at example.com that must be responsible authorizing the use of names in the example.com domain. Therefore, the credentials needed to prove this authorization must ultimately derive from the domain owner: either a user agent gives requests to the domain name owner in order for them to be signed by the domain owner's credentials, or the user agent must possess credentials that prove in some fashion that the domain owner has given the user agent the right to a name.

The situation is however more complicated for telephone numbers. Authority over telephone numbers does not correspond directly to Internet domains. While a user could register at a SIP domain with a username that corresponds to a telephone number, any connection between the administrator of that domain and the assignment of telephone numbers is not reflected on the Internet. Telephone numbers do not share the domain-scope property described above, as they are dialed without any domain component. This document thus assumes the existence of a separate means of establishing authority over telephone numbers, for cases where the telephone number is the identity of the user. As with SIP URIs, the necessary credentials to prove authority for a name might reside either in the endpoint or at some intermediary.

This document specifies a means of sharing a cryptographic assurance of end-user SIP identity in an interdomain or intradomain context that is based on the authentication service adding a SIP header, the Identity header. In order to assist in the validation of this assurance, this specification also describes an Identity-Info header that can be used by the recipient of a request to recover the credentials of the signer. Note that the scope of this document is limited to providing this identity assurance for SIP requests; solving this problem for SIP responses is outside the scope of this work.

This specification allows either a user agent or a proxy server to provide identity services and to verify identities. To maximize end-to-end security, it is obviously preferable for end-users to acquire their own credentials; if they do, they can act as an authentication service. However, end-user credentials may be neither practical nor affordable, given the potentially large number of SIP user agents (phones, PCs, laptops, PDAs, gaming devices) that may be employed by a single user. In such environments, synchronizing keying material across multiple devices may be very complex and requires quite a good deal of additional endpoint behavior. Managing several credentials for the various devices could also be burdensome. This trade-off needs to be understood by implementers of this specification.

3.1. Intermediary Authentication Services

In cases where a user agent does not possess its own credentials to sign an Identity header, the user agent must send its request through an intermediary that will provide a signed Identity header based on the contents of the request. This requires, among other things, that intermediaries have some means of authenticating the user agents sending requests.

All RFC 3261 [RFC3261] compliant user agents support Digest authentication, which utilizes a shared secret, as a means for authenticating themselves to a SIP registrar. Registration allows a user agent to express that it is an appropriate entity to which requests should be sent for a particular SIP AoR URI (e.g., 'sip:alice@atlanta.example.com'). For such SIP URIs, by the definition of identity used in this document, registration proves the identity of the user to a registrar. Similar checks might be performed for telephone numbers as identities. This is of course only one manner in which a domain might determine how a particular user is authorized to populate the From header field; as an aside, for other sorts of URIs in the From (like anonymous URIs), other authorization policies would apply.

RFC 3261 [RFC3261] already describes an intermediary architecture very similar to the one proposed in this document in Section 26.3.2.2, in which a user agent authenticates itself to a local proxy server, which in turn authenticates itself to a remote proxy server via mutual TLS, creating a two-link chain of transitive authentication between the originator and the remote domain. While this works well in some architectures, there are a few respects in which this is impractical. For one, transitive trust is inherently weaker than an assertion that can be validated end-to-end. It is possible for SIP requests to cross multiple intermediaries in separate administrative domains, in which case transitive trust becomes even less compelling.

One solution to this problem is to use 'trusted' SIP intermediaries that assert an identity for users in the form of a privileged SIP header. A mechanism for doing so (with the P-Asserted-Identity header) is given in [12]. However, this solution allows only hop-by-hop trust between intermediaries, not end-to-end cryptographic authentication, and it assumes a managed network of nodes with strict mutual trust relationships, an assumption that is incompatible with widespread Internet deployment.

4. Overview of Operations

This section provides an informative (non-normative) high-level overview of the mechanisms described in this document.

Imagine the case where Alice, who has the home proxy of example.com and the address-of-record sip:alice@example.com, wants to communicate with sip:bob@example.org.

Alice generates an INVITE and places her identity in the From header field of the request. She then sends an INVITE over TLS to an authentication service proxy for her domain.

The authentication service authenticates Alice (possibly by sending a Digest authentication challenge) and validates that she is authorized to assert the identity that is populated in the From header field. This value may be Alice's AoR, or it may be some other value that the proxy server has authority over, such as a telephone number. It then computes a hash over some particular headers, including the From header field (and, optionally the body) in the message. This hash is signed with the appropriate credential (example.com, in the sip:alice@example.com case) and inserted in a new header field in the SIP message, the 'Identity' header.

The proxy, as the holder of the private key for its domain, is asserting that the originator of this request has been authenticated and that she is authorized to claim the identity (the SIP address-of-record) that appears in the From header field. The proxy also inserts a companion header field, Identity-Info, that tells Bob how to acquire keying material necessary to validate its credentials, if he doesn't already have it.

When Bob's domain receives the request, it verifies the signature provided in the Identity header, and thus can validate that the authority over the identity in the From header field authenticated the user, and permitted the user to assert that From header field value. This same validation operation may be performed by Bob's user agent server (UAS).

5. Signature Generation and Validation

5.1. Authentication Service Behavior

This document defines a role for SIP entities called an authentication service. The authentication service role can be instantiated by a proxy server or a user agent. Any entity that instantiates the authentication service role MUST possess the private key of one or more credentials that can be used to sign for a domain or a telephone number (see Section 6.1). Intermediaries that instantiate this role MUST be capable of authenticating one or more

SIP users who can register for that identity. Commonly, this role will be instantiated by a proxy server, since these entities are more likely to have a static hostname, hold corresponding credentials, and have access to SIP registrar capabilities that allow them to authenticate users. It is also possible that the authentication service role might be instantiated by an entity that acts as a redirect server, but that is left as a topic for future work.

SIP entities that act as an authentication service MUST add a Date header field to SIP requests if one is not already present (see Section 10 for information on how the Date header field assists verifiers). Similarly, authentication services MUST add a Content-Length header field to SIP requests if one is not already present; this can help verifiers to double-check that they are hashing exactly as many bytes of message-body as the authentication service when they verify the message.

Entities instantiating the authentication service role perform the following steps, in order, to generate an Identity header for a SIP request:

Step 1:

The authentication service MUST extract the identity of the sender from the request. The authentication service takes this value from the From header field; this AoR will be referred to here as the 'identity field'. If the identity field contains a SIP or SIP Secure (SIPS) URI, and the user portion is not a telephone number, the authentication service MUST extract the hostname portion of the identity field and compare it to the domain(s) for which it is responsible (following the procedures in RFC 3261 [RFC3261], Section 16.4), used by a proxy server to determine the domain(s) for which it is responsible). If the identity field uses the TEL URI scheme, or the identity field is a SIP or SIPS URI with a telephone number in the user portion, the authentication service determines whether or not it is responsible for this telephone number; see Section 7 for more information. If the authentication service is not authoritative for the identity in question, it SHOULD process and forward the request normally, but it MUST NOT add an Identity header; see below for more information on authentication service handling of an existing Identity header.

Step 2:

The authentication service MUST determine whether or not the sender of the request is authorized to claim the identity given in the identity field. In order to do so, the authentication service MUST authenticate the sender of the message. Some possible ways in which this authentication might be performed include:

If the authentication service is instantiated by a SIP intermediary (proxy server), it may challenge the request with a 407 response code using the Digest authentication scheme (or viewing a Proxy-Authentication header sent in the request, which was sent in anticipation of a challenge using cached credentials, as described in RFC 3261 [RFC3261], Section 22.3). Note that if that proxy server is maintaining a TLS connection with the client over which the client had previously authenticated itself using Digest authentication, the identity value obtained from that previous authentication step can be reused without an additional Digest challenge.

If the authentication service is instantiated by a SIP user agent, a user agent can be said to authenticate its user on the grounds that the user can provision the user agent with the private key of the credential, or preferably by providing a password that unlocks said private key.

Authorization of the use of a particular username or telephone number in the user part of the From header field is a matter of local policy for the authentication service, see Section 6.1 for more information.

Note that this check is performed on the addr-spec in the From header field (e.g., the URI of the sender, like 'sip:alice@atlanta.example.com'); it does not convert the display-name portion of the From header field (e.g., 'Alice Atlanta'). Authentication services MAY check and validate the display-name as well, and compare it to a list of acceptable display-names that may be used by the sender; if the display-name does not meet policy constraints, the authentication service MUST return a 403 response code. The reason phrase should indicate the nature of the problem; for example, "Inappropriate Display Name". However, the display-name is not always present, and in many environments the requisite operational procedures for display-name validation may not exist. For more information, see Section 13.2.

Step 3:

The authentication service SHOULD ensure that any preexisting Date header in the request is accurate. Local policy can dictate precisely how accurate the Date must be; a RECOMMENDED maximum discrepancy of ten minutes will ensure that the request is unlikely

to upset any verifiers. If the Date header contains a time different by more than ten minutes from the current time noted by the authentication service, the authentication service SHOULD reject the request. This behavior is not mandatory because a user agent client (UAC) could only exploit the Date header in order to cause a request to fail verification; the Identity header is not intended to provide a source of non-repudiation or a perfect record of when messages are processed. Finally, the authentication service MUST verify that the Date header falls within the validity period of its credential. For more information on the security properties associated with the Date header field value, see Section 10.

[TBD: Should consider a lower threshold than ten minutes? With the removal of other elements from the sig, that's a lot of leeway.]

Step 4:

The authentication service MAY form an identity-reliance signature and add an Identity-Reliance header to the request containing this signature. The Identity-Reliance header provides body security properties that are useful for non-INVITE transactions, and in environments where body security of INVITE transactions is necessary. Details on the generation of this header is provided in Section 10.

Step 5:

The authentication service MUST form the identity signature and add an Identity header to the request containing this signature. After the Identity header has been added to the request, the authentication service MUST also add an Identity-Info header. The Identity-Info header contains a URI from which its credential can be acquired; see Section 6.3 for more on credential acquisition. Details on the syntax of both of these headers are provided in Section 10.

Finally, the authentication service MUST forward the message normally.

5.1.1. Identity within a Dialog and Retargeting

Retargeting is broadly defined as the alteration of the Request-URI by intermediaries. More specifically, retargeting supplants the original target URI with one that corresponds to a different user, a user that is not authorized to register under the original target URI. By this definition, retargeting does not include translation of the Request-URI to a contact address of an endpoint that has registered under the original target URI, for example.

When a dialog-forming request is retargeted, this can cause a few wrinkles for the Identity mechanism when it is applied to requests sent in the backwards direction within a dialog. This section provides some non-normative considerations related to this case.

When a request is retargeted, it may reach a SIP endpoint whose user is not identified by the URI designated in the To header field value. The value in the To header field of a dialog-forming request is used as the From header field of requests sent in the backwards direction during the dialog, and is accordingly the header that would be signed by an authentication service for requests sent in the backwards direction. In retargeting cases, if the URI in the From header does not identify the sender of the request in the backwards direction, then clearly it would be inappropriate to provide an Identity signature over that From header. As specified above, if the authentication service is not responsible for the domain in the From header field of the request, it MUST NOT add an Identity header to the request, and it should process/forward the request normally.

Any means of anticipating retargeting, and so on, is outside the scope of this document, and likely to have equal applicability to response identity as it does to requests in the backwards direction within a dialog. Consequently, no special guidance is given for implementers here regarding the 'connected party' problem; authentication service behavior is unchanged if retargeting has occurred for a dialog-forming request. Ultimately, the authentication service provides an Identity header for requests in the backwards dialog when the user is authorized to assert the identity given in the From header field, and if they are not, an Identity header is not provided.

For further information on the problems of response identity and the potential solution spaces, see [15].

5.2. Verifier Behavior

This document introduces a new logical role for SIP entities called a verification service or verifier. When a verifier receives a SIP message containing an Identity header, it may inspect the signature to verify the identity of the sender of the message. Typically, the results of a verification are provided as input to an authorization process that is outside the scope of this document. If an Identity header is not present in a request, and one is required by local policy (for example, based on a per-sending-domain policy, or a per-sending-user policy), then a 428 'Use Identity Header' response MUST be sent.

In order to verify the identity of the sender of a message, an entity acting as a verifier MUST perform the following steps, in the order here specified.

Step 1:

In order to determine whether the signature for the URI in the From header field value should be over the entire URI or just a canonicalized telephone number, the verification service must follow the process described in Section 7. That section also describes the procedures the verification service must follow to determine if the signer is authoritative for a telephone number. For domains, the verifier MUST follow the process described in Section 13.4 to determine if the signer is authoritative for the URI in the From header field.

Step 2:

The verifier must first ensure that it possesses the proper keying material to validate the signature in the Identity header field. See Section 6.2 for more information on these procedures.

Step 3:

The verifier MUST verify the signature in the Identity header field, following the procedures for generating the hashed digest-string described in Section 10. If a verifier determines that the signature on the message does not correspond to the reconstructed digest-string, then a 438 'Invalid Identity Header' response MUST be returned.

Step 4:

If the request contains an Identity-Reliance header, the verifier SHOULD verify the signature in the Identity-Reliance header field, following the procedures for generating the hashed reliance-digest-string described in Section 10. If a verifier determines that the signature on the message does not correspond to the reconstructed digest-string, then a 438 'Invalid Identity Header' response SHOULD be returned.

Step 5:

The verifier MUST validate the Date header in the manner described in Section 13.1; recipients that wish to verify Identity signatures MUST support all of the operations described there. It must furthermore ensure that the value of the Date header falls within the validity period of the certificate whose corresponding private key was used to sign the Identity header.

6. Credentials

SIP entities cannot reliably predict where SIP requests will terminate. When choosing a credential scheme for deployments of this specification, it is therefore essential that the trust anchor(s) for credentials be widely trusted, or that deployments restrict the use of this mechanism to environments where the reliance on particular trust anchors is assured by business arrangements or similar constraints.

For more on the use of certificates for domain names as a credential system, see Section 13.4.

6.1. Credential Use by the Authentication Service

In order to act as an authentication service, a SIP entity must have access to the private keying material of one or more credentials that cover URIs, domain names or telephone numbers. These credentials may represent authority over only a single name (such as alice@example.com), an entire domain (such as example.com), or potentially a set of domains. Similarly, a credential may represent authority over a single telephone number or a range of telephone numbers. The way that the scope of a credential is expressed is specific to the credential mechanism.

Authorization of the use of a particular username or telephone number in the user part of the From header field is a matter of local policy for the authentication service, one that depends greatly on the manner in which authentication is performed. For non-telephone number user parts, one policy might be as follows: the username given in the 'username' parameter of the Proxy-Authorization header MUST correspond exactly to the username in the From header field of the SIP message. However, there are many cases in which this is too limiting or inappropriate; a realm might use 'username' parameters in Proxy-Authorization that do not correspond to the user-portion of SIP From headers, or a user might manage multiple accounts in the same administrative domain. In this latter case, a domain might maintain a mapping between the values in the 'username' parameter of Proxy-Authorization and a set of one or more SIP URIs that might legitimately be asserted for that 'username'. For example, the username can correspond to the 'private identity' as defined in Third

Generation Partnership Project (3GPP), in which case the From header field can contain any one of the public identities associated with this private identity. In this instance, another policy might be as follows: the URI in the From header field MUST correspond exactly to one of the mapped URIs associated with the 'username' given in the Proxy-Authorization header. This is a suitable approach for telephone numbers in particular. Various exceptions to such policies might arise for cases like anonymity; if the AoR asserted in the From header field uses a form like 'sip:anonymous@example.com', then the 'example.com' proxy should authenticate that the user is a valid user in the domain and insert the signature over the From header field as usual.

6.2. Credential Use by the Verification Service

In order to act as a verification service, a SIP entity must have a way to acquire and retain credentials for authorities over particular URIs, domain names and/or telephone numbers. The Identity-Info header (as described in the next section) is supported by all verification service implementations to create a baseline means of credential acquisition. Provided that the credential used to sign a message is not previously known to the verifier, SIP entities SHOULD discover this credential by dereferencing the Identity-Info header, unless they have some more efficient implementation-specific way of acquiring certificates. If the URI scheme in the Identity-Info header cannot be dereferenced, then a 436 'Bad Identity-Info' response MUST be returned.

In the case the credential is a certificate, the verifier processes this certificate in the usual ways, including checking that it has not expired, that the chain is valid back to a trusted certificate authority (CA), and that it does not appear on revocation lists. Once the certificate is acquired, it MUST be validated following the procedures in RFC 3280 [RFC3280]. If the certificate cannot be validated (it is self-signed and untrusted, or signed by an untrusted or unknown certificate authority, expired, or revoked), the verifier MUST send a 437 'Unsupported Certificate' response.

Verification service implementations supporting this specification SHOULD have some means of retaining credentials (in accordance with normal practices for credential lifetimes and revocation) in order to prevent themselves from needlessly downloading the same credential every time a request from the same identity is received. Credentials cached in this manner SHOULD be indexed by their scope, or the URI given in the Identity-Info header field value.

6.3. Handling Identity-Info URIs

A URI in an Identity-Info header MUST contain a URI which dereferences to a resource containing the credential used by the authentication service to sign a request. Much as is the case with the trust anchor(s) required for deployments of this specification, it is essential that a URI in the Identity-Info header be dereferencable by any entity that can receive the request. For common cases, this means that the URI must be dereferencable by any entity on the public Internet. In constrained deployment environments, a service private to the environment might be used instead.

Beyond providing a means of accessing credentials for an identity, the Identity-Info header further services a means of differentiating which particular credential was used to sign a request, when there are potentially multiple authorities eligible to sign. For example, imagine a case where a domain implements the authentication service role for example.com, and a user agent belonging to Alice has acquired a credential for alice@example.com. Either would be eligible to sign a SIP request from alice@example.com. Verification services however need a means to differentiate which one performed the signature. The Identity-Info header performs that function.

All implementations of this specification MUST support the use of HTTP and HTTPS URIs in the Identity-Info header. Such HTTP and HTTPS URIs MUST follow the conventions of RFC 2585 [RFC2585], and for those URIs the indicated resource MUST be of the form 'application/pkix-cert' described in that specification. Note that this introduces key lifecycle management concerns; were a domain to change the key available at the Identity-Info URI before a verifier evaluates a request signed by an authentication service, this would cause obvious verifier failures. When a rollover occurs, authentication services SHOULD thus provide new Identity-Info URIs for each new certificate, and SHOULD continue to make older key acquisition URIs available for a duration longer than the plausible lifetime of a SIP message (an hour would most likely suffice).

Beyond HTTP, implementations may support any of several alternative mechanism for acquiring credentials. When implemented as part of a user agent, for example, an authentication service might include its credential as an additional MIME body in the SIP request, and refer to the certificate with a CID URI (per [RFC2392]). Uses of SIP outside of the request transaction may be suitable for transmitting certificates in some environments, such as through a SUBSCRIBE/NOTIFY exchange. As DANE deployment increases with the widespread adoption of DNSSEC, implementations may want to rely on keying material stored in the DNS. The Identity-Info headers may use the DNS URL scheme to indicate keys in the DNS.

[TBD: Should we add some kind of hash or similar indication to the Identity-Info header to make it easier for verifiers to ascertain that they already possess a credential without dereferencing the URI?]

7. Identity and Telephone Numbers

Since many SIP applications provide a Voice over IP (VoIP) service, telephone numbers are commonly used as identities in SIP deployments. In order for telephone numbers to be used with the mechanism described in this document, authentication services must enroll with an authority that issues credentials for telephone numbers or telephone number ranges, and verification services must trust the authority employed by the authentication service that signs a request.

Given the existence of such authorities, authentication and verification services must furthermore identify when a request should be signed by an authority for a telephone number, and when it should be signed by an authority for a domain. Telephone numbers most commonly appear in SIP requests in the username portion of a SIP URI (e.g., 'sip:+17005551008@chicago.example.com;user=phone'). The user part of that URI conforms to the syntax of the TEL URI scheme (RFC 3966 [RFC3966]). It is also possible for a TEL URI to appear in the SIP To or From header field outside the context of a SIP or SIPS URI (e.g., 'tel:+17005551008'). In both of these cases, it's clear that the signer must have authority over the telephone number, not the domain name of the SIP URI. It is also possible, however, for requests to contain a URI like 'sip:7005551000@chicago.example.com'. It may be non-trivial for a service to ascertain in this case whether the URI contains a telephone number or not.

To address this problem, the authentication service and verification service both must perform the following canonicalization procedure on any SIP URI they inspect which contains a wholly numeric user part. [TBD: the algorithm] If the result of this procedure forms a complete telephone number, that number is used for the purpose of creating and signing the digest-string at the authentication service and verification service. If the result does not form a complete telephone number, the authentication service and verification service should treat the entire URI as a SIP URI, and apply a domain signature per the procedures in Section 13.4.

This specification assumes that UACs will have an appropriate means to discover an authentication service that can sign with a telephone number certificate corresponding to the UAC's telephone number. Most likely, this information will simply be provisioned in UACs.

Certificates that prove authority over telephone numbers should contain the telephone number, or number range, in the [TBD] field of the certificate. Verification services must compare the canonicalized telephone number to the contents of the [TBD] field in order to establish that the proper authority has signed the request. [TBD: This would refer to an external specification, most likely]

In the longer term, it is possible that some directory or other discovery mechanism may provide a way to determine which administrative domain is responsible for a telephone number, and this may aid in the signing and verification of SIP identities that contain telephone numbers. This is a subject for future work.

8. Considerations for User Agents

This mechanism can be applied opportunistically to existing SIP deployments; accordingly, it requires no change to SIP user agent behavior in order for it to be effective. However, because this mechanism does not provide integrity protection between the UAC and the authentication service, a UAC SHOULD implement some means of providing this integrity. TLS would be one such mechanism, which is attractive because it MUST be supported by SIP proxy servers, but is potentially problematic because it is a hop-by-hop mechanism. See Section 13.3 for more information about securing the channel between the UAC and the authentication service.

When a UAC sends a request, it MUST accurately populate the From header field with a value corresponding to an identity that it believes it is authorized to claim. In a request, it MUST set the URI portion of its From header to match a SIP, SIPS, or TEL URI AoR that it is authorized to use in the domain (including anonymous URIs, as described in RFC 3323 [RFC3323]).

Note that this document defines a number of new 4xx response codes. If user agents support these response codes, they will be able to respond intelligently to Identity-based error conditions.

The UAC MUST also be capable of sending requests, including mid-call requests, through an 'outbound' proxy (the authentication service). The best way to accomplish this is using pre-loaded Route headers and loose routing. For a given domain, if an entity that can instantiate the authentication service role is not in the path of dialog-forming requests, identity for mid-dialog requests in the backwards direction cannot be provided.

As a recipient of a request, a user agent that can verify signed identities should also support an appropriate user interface to render the validity of identity to a user. User agent

implementations SHOULD differentiate signed From header field values from unsigned From header field values when rendering to an end-user the identity of the sender of a request.

9. Considerations for Proxy Servers

Domain policy may require proxy servers to inspect and verify the identity provided in SIP requests. A proxy server may wish to ascertain the identity of the sender of the message to provide spam prevention or call control services. Even if a proxy server does not act as an verification service, it MAY validate the Identity header before it makes a forwarding decision for a request. Compliant proxy servers MUST NOT remove or modify an existing Identity or Identity-Info header in a request.

10. Header Syntax

This document specifies three SIP headers: Identity, Identity-Reliance and Identity- Info. Each of these headers can appear only once in a SIP request; Identity-Reliance is OPTIONAL, while Identity and Identity-Info are REQUIRED for securing requests with this specification. The grammar for these three headers is (following the ABNF [6] in RFC 3261 [1]):

```
   Identity = "Identity" HCOLON signed-identity-digest
   signed-identity-digest = LDQUOT 32LHEX RDQUOT

   Identity-Reliance = "Identity-Reliance" HCOLON signed-identity-reliance-
digest
   signed-identity-reliance-digest = LDQUOT 32LHEX RDQUOT

   Identity-Info = "Identity-Info" HCOLON ident-info
                   *( SEMI ident-info-params )
   ident-info = LAQUOT absoluteURI RAQUOT
   ident-info-params = ident-info-alg / ident-info-extension
   ident-info-alg = "alg" EQUAL token
   ident-info-extension = generic-param
```

[TBD: The version has the Identity-Reliance header covered under the Identity signature. It is also possible to do this the other way around, where the base Identity signature is generated first, and Identity-Reliance would cover both the Identity header and the body. This is a trade-off of whether the authentication service should decide whether Identity-Reliance is needed or if the verification service should decide. These have different properties, and some investigation would be needed to decide between them.]

The signed-identity-reliance-digest is a signed hash of a canonical string generated from certain components of a SIP request. Creating this hash and the Identity-Reliance header field to contain it is OPTIONAL, and its usage is a matter of policy for authentication services. To create the contents of the signed-identity-digest, the following element of a SIP message MUST be placed in a bit-exact string:

> The body content of the message with the bits exactly as they are in the message (in the ABNF for SIP, the message-body). This includes all components of multipart message bodies. Note that the message-body does NOT include the CRLF separating the SIP headers from the message-body, but does include everything that follows that CRLF.

[TBD: Explore alternatives to including the whole body for INVITE requests]

The signed-identity-digest is a signed hash of a canonical string generated from certain components of a SIP request. To create the contents of the signed-identity-digest, the following elements of a SIP message MUST be placed in a bit-exact string in the order specified here, separated by a vertical line, "|" or %x7C, character:

> First, the identity. If the user part of the AoR in the From header field of the request contains a telephone number, then the canonicalization of that number goes into the first slot (see Section 7). Otherwise, the first slot contains the AoR of the UA sending the message, or addr-spec of the From header field.
>
> Second, the target. If the user part of the AoR in the To header field of the request contains a telephone number, then the canonicalization of that number goes into the second slot (see Section 7). Otherwise, the second slot contains the addr-spec component of the To header field, which is the AoR to which the request is being sent.
>
> Third, the request method.
>
> Fourth, the Date header field, with exactly one space each for each SP and the weekday and month items case set as shown in BNF in RFC 3261 [RFC3261]. RFC 3261 specifies that the BNF for weekday and month is a choice amongst a set of tokens. The RFC 2234 [RFC2234] rules for the BNF specify that tokens are case sensitive. However, when used to construct the canonical string defined here, the first letter of each week and month MUST be capitalized, and the remaining two letters must be lowercase. This matches the capitalization provided in the definition of each

token. All requests that use the Identity mechanism MUST contain a Date header.

Fifth, the Identity-Reliance header field value, if there is an Identity-Reliance field in the request. If the message has no body, or no Identity-Reliance header, then the fifth slot will be empty, and the final "|" will not be followed by any additional characters.

[TBD: Should there be a special case for security parameters that would appear in SDP?]

For more information on the security properties of these headers, and why their inclusion mitigates replay attacks, see Section 13 and [RFC3893]. The precise formulation of this digest-string is, therefore (following the ABNF[RFC4234] in RFC 3261 [RFC3261]):

```
digest-string = addr-spec / tn-spec "|" addr-spec / tn-spec "|"
                Method "|" SIP-date "|" [ signed-identity-reliance-digest ]
```

For the definition of 'tn-spec' see Section 7.

After the digest-string or reliance-digest-string is formed, each MUST be hashed and signed with the certificate of authority over the identity. The hashing and signing algorithm is specified by the 'alg' parameter of the Identity-Info header (see below for more information on Identity-Info header parameters). This document defines only one value for the 'alg' parameter: 'rsa-sha1'; further values MUST be defined in a Standards Track RFC, see Section 14.7 for more information. All implementations of this specification MUST support 'rsa-sha1'. When the 'rsa-sha1' algorithm is specified in the 'alg' parameter of Identity-Info, the hash and signature MUST be generated as follows: compute the results of signing this string with sha1WithRSAEncryption as described in RFC 3370 [RFC3370] and base64 encode the results as specified in RFC 3548 [RFC3548]. A 1024-bit or longer RSA key MUST be used. The result of the digest-string hash is placed in the Identity header field; the optional reliance-digest-string hash goes in the Identity-Reliance header. For detailed examples of the usage of this algorithm, see Section 11.

The 'absoluteURI' portion of the Identity-Info header MUST contain a URI; see Section 6.3 for more on choosing how to advertise credentials through Identity-Info.

The Identity-Info header field MUST contain an 'alg' parameter. No other parameters are defined for the Identity-Info header in this document. Future Standards Track RFCs may define additional Identity-Info header parameters.

This document adds the following entries to Table 2 of RFC 3261 [RFC3261] (this repeats the registrations of RFC4474):

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Identity	R	a	o	o	-	o	o	o

SUB	NOT	REF	INF	UPD	PRA
o	o	o	o	o	o

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Identity-Info	R	a	o	o	-	o	o	o

SUB	NOT	REF	INF	UPD	PRA
o	o	o	o	o	o

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Identity-Reliance	R	a	o	o	-	o	o	o

SUB	NOT	REF	INF	UPD	PRA
o	o	o	o	o	o

Note, in the table above, that this mechanism does not protect the CANCEL method. The CANCEL method cannot be challenged, because it is hop-by-hop, and accordingly authentication service behavior for CANCEL would be significantly limited. The Identity and Identity-Info header MUST NOT appear in CANCEL. Note as well that the use of Identity with REGISTER is consequently a subject for future study, although it is left as optional here for forward-compatibility reasons.

11. Compliance Tests and Examples

[TBD: Need to fix examples for RFC4474bis]

The examples in this section illustrate the use of the Identity header in the context of a SIP transaction. Implementers are advised to verify their compliance with the specification against the following criteria:

Implementations of the authentication service role MUST generate identical base64 identity strings to the ones shown in the Identity headers in these examples when presented with the source message and utilizing the appropriate supplied private key for the domain in question.

Implementations of the verifier role MUST correctly validate the given messages containing the Identity header when utilizing the supplied certificates (with the caveat about self-signed certificates below).

Note that the following examples use self-signed certificates, rather than certificates issued by a recognized certificate authority. The use of self-signed certificates for this mechanism is NOT RECOMMENDED, and it appears here only for illustrative purposes. Therefore, in compliance testing, implementations of verifiers SHOULD generate appropriate warnings about the use of self-signed certificates. Also, the example certificates in this section have placed their domain name subject in the subjectAltName field; in practice, certificate authorities may place domain names in other locations in the certificate (see Section 13.4 for more information).

Note that all examples in this section use the 'rsa-sha1' algorithm.

Bit-exact reference files for these messages and their various transformations are supplied in Appendix B.

11.1. Identity-Info with a Singlepart MIME body

Consider the following private key and certificate pair assigned to 'atlanta.example.com' (rendered in OpenSSL format).

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQDPPMBtHVoPkXV+Z6jq1LsgfTELVWpy2BVUffJMPH06LL0cJSQO
aIeVzIojzWtpauB7IylZKlAjB5f429tRuoUiedCwMLKblWAqZt6eHWpCNZJ7lONc
IEwnmh2nAccKk83Lp/VH3tgAS/43DQoX2sndnYh+g8522Pzwg7EGWspzzwIDAQAB
AoGBAK0W3tnEFD7AjVQAnJNXDtx59Aa1Vu2JEXe6oi+OrkFysJjbZJwsLmKtrgtt
PXOU8t2mZpi0wK4hX4tZhntiwGKkUPC3h9Bjp+GerifP341RMyMO+6fPgjqOzUDw
+rPjjMpwD7AkcEcqDgbTrZnWv/QnCSaaF3xkUGfFkLx5OKcRAkEA7UxnsE8XaT30
```

```
tP/UUc51gNk2KGKgxQQTHopBcew9yfeCRFhvdL7jpaGatEi5iZwGGQQDVOVHUN1H
0YLpHQjRowJBAN+R2bvA/Nimq464ZgnelEDPqaEAZWaD3kOfhS9+vL7oqES+u5E0
J7kXb7ZkiSVUg9XU/8PxMKx/DAz0dUmOL+UCQH8C9ETUMI2uEbqHbBdVUGNk364C
DFcndSxVh+34KqJdjiYSx6VPPv26X9m7S0OydTkSgs3/4ooPxo8HaMqXm80CQB+r
xbB3UlpOohcBwFK9mTrlMB6Cs9ql66KgwnlL9ukEhHHYozGatdXeoBCyhUsogdSU
6/aSAFcvWEGtj7/vyJECQQCCS1lKgEXoNQPqONalvYhyyMZRXFLdD4gbwRPK1uXK
Ypk3CkfFzOyfjeLcGPxXzq2qzuHzGTDxZ9PAepwX4RSk
-----END RSA PRIVATE KEY-----


-----BEGIN CERTIFICATE-----
MIIC3TCCAkagAwIBAgIBADANBgkqhkiG9w0BAQUFADBZMQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCR0ExEDAOBgNVBAcMB0F0bGFudGExDTALBgNVBAoMBElFVEYxHDAa
BgNVBAMME2F0bGFudGEuZXhhbXBsZS5jb20wHhcNMDUxMDI0MDYzNjA2WhcNMDYx
MDI0MDYzNjA2WjBZMQswCQYDVQQGEwJVUzELMAkGA1UECAwCR0ExEDAOBgNVBAcM
B0F0bGFudGExDTALBgNVBAoMBElFVEYxHDAaBgNVBAMME2F0bGFudGEuZXhhbXBs
ZS5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAM88wG0dWg+RdX5nqOrU
uyB9MQtVanLYFVR98kw8fTosvRwlJA5oh5XMiiPNa2lq4HsjKVkqUCMHl/jb21G6
hSJ50LAwspuVYCpm3p4dakI1knuU41wgTCeaHacBxwqTzcun9Ufe2ABL/jcNChfa
yd2diH6DznbY/PCDsQZaynPPAgMBAAGjgbQwgbEwHQYDVR0OBBYEFNmU/MrbVYcE
KDr/20WISrG1j1rNMIGBBgNVHSMEejB4gBTZlPzK21WHBCg6/9tFiEqxtY9azaFd
pFswWTELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAkdBMRAwDgYDVQQHDAdBdGxhbnRh
MQ0wCwYDVQQKDARJRVRGMRwwGgYDVQQDDBNhdGxhbnRhLmV4YW1wbGUuY29tggEA
MAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQEFBQADgYEADdQYtswBDmTSTq0mt211
7alm/XGFrb2zdbU0vorxRdOZ04qMyrIpXG1LEmnEOgcocyrXRBvq5p6WbZAcEQk0
DsE3Ve0Nc8x9nmvljW7GsMGFCnCuo4ODTf/1lGdVr9DeCzcj10YUQ3MRemDMXhY2
CtDisLWl7SXOORcZAi1oU9w=
 -----END CERTIFICATE-----
```

A user of atlanta.example.com, Alice, wants to send an INVITE to bob@biloxi.example.org. She therefore creates the following INVITE request, which she forwards to the atlanta.example.org proxy server that instantiates the authentication service role:

```
INVITE sip:bob@biloxi.example.org SIP/2.0
Via: SIP/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bKnashds8
To: Bob <sip:bob@biloxi.example.org>
From: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Date: Thu, 21 Feb 2002 13:02:03 GMT
Contact: <sip:alice@pc33.atlanta.example.com>
Content-Type: application/sdp
Content-Length: 147

v=0
```

```
o=UserA 2890844526 2890844526 IN IP4 pc33.atlanta.example.com
s=Session SDP
c=IN IP4 pc33.atlanta.example.com
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

When the authentication service receives the INVITE, it authenticates Alice by sending a 407 response. As a result, Alice adds an Authorization header to her request, and resends to the atlanta.example.com authentication service. Now that the service is sure of Alice's identity, it calculates an Identity header for the request. The canonical string over which the identity signature will be generated is the following (note that the first line wraps because of RFC editorial conventions):

```
sip:alice@atlanta.example.com|sip:bob@biloxi.example.org|
INVITE|Thu, 21 Feb 2002 13:02:03 GMT|
```

The resulting signature (sha1WithRsaEncryption) using the private RSA key given above, with base64 encoding, is the following:

```
ZYNBbHC00VMZr2kZt6VmCvPonWJMGvQTBDqghoWeLxJfzB2a1pxAr3VgrB0SsSAa
ifsRdiOPoQZYOy2wrVghuhcsMbHWUSFxI6p6q5TOQXHMmz6uEo3svJsSH49thyGn
FVcnyaZ++yRlBYYQTLqWzJ+KVhPKbfU/pryhVn9Yc6U=
```

Accordingly, the atlanta.example.com authentication service will create an Identity header containing that base64 signature string (175 bytes). It will also add an HTTPS URL where its certificate is made available. With those two headers added, the message looks like the following:

```
INVITE sip:bob@biloxi.example.org SIP/2.0
Via: SIP/2.0/TLS pc33.atlanta.example.com;branch=z9hG4bKnashds8
To: Bob <sip:bob@biloxi.example.org>
From: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Date: Thu, 21 Feb 2002 13:02:03 GMT
Contact: <sip:alice@pc33.atlanta.example.com>
Identity:
  "ZYNBbHC00VMZr2kZt6VmCvPonWJMGvQTBDqghoWeLxJfzB2a1pxAr3VgrB0SsSAa
   ifsRdiOPoQZYOy2wrVghuhcsMbHWUSFxI6p6q5TOQXHMmz6uEo3svJsSH49thyGn
   FVcnyaZ++yRlBYYQTLqWzJ+KVhPKbfU/pryhVn9Yc6U="
```

```
Identity-Info: <https://atlanta.example.com/atlanta.cer>;alg=rsa-sha1
Content-Type: application/sdp
Content-Length: 147
 v=0
o=UserA 2890844526 2890844526 IN IP4 pc33.atlanta.example.com
s=Session SDP
c=IN IP4 pc33.atlanta.example.com
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

atlanta.example.com then forwards the request normally. When Bob receives the request, if he does not already know the certificate of atlanta.example.com, he dereferences the URL in the Identity-Info header to acquire the certificate. Bob then generates the same canonical string given above, from the same headers of the SIP request. Using this canonical string, the signed digest in the Identity header, and the certificate discovered by dereferencing the

Identity-Info header, Bob can verify that the given set of headers and the message body have not been modified.

11.2. Identity for a Request with No MIME Body or Contact

Consider the following private key and certificate pair assigned to "biloxi.example.org".

```
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQC/obBYLRMPjskrAqWOiGPAUxI3/m2ti7ix4caqCTAuFX5cLegQ
7nmquLOHfIhxVIqT2f06UA0lOo2NVofK9G7MTkVbVNiyAlLYUDEj7XWLDICf3ZHL
6Fr/+CF7wrQ9r4kv7XiJKxodVCCd/DhCT9Gp+VDoe8HymqOW/KsneriyIwIDAQAB
AoGBAJ7fsFIKXKkjWgj8ksGOthS3Sn19xPSCyEdBxfEm2Pj7/Nzzeli/PcOaic0k
JALBcnqN2fHEeIGK/9xUBxTufgQYVJqvyHERs6rXX/iT4Ynm9t1905EiQ9ZpHsrI
/AMMUYA1QrGgAIHvZLVLzq+9KLDEZ+HQbuCLJXF+6bl0Eb5BAkEA636oMANp0Qa3
mYWEQ2utmGsYxkXSfyBb18TCOwCty0ndBR24zyOJF2NbZS98Lz+Ga25hfIGw/JHK
nD9bOE88UwJBANBRSpd4bmS+m48R/13tRESAtHqydNinX0kS/RhwHr7mkHTU3k/M
FxQtx34I3GKzaZxMn0A66KS9v/SHdnF+ePECQQCGe7QshyZ8uitLPtZDclCWhEKH
qAQHmUEZvUF2VHLrbukLLOgHUrHNa24cILv4d3yaCVUetymNcuyTwhKj24wFAkAO
z/jx1EplN3hwL+NsllZoWI58uvu7/Aq2c3czqaVGBbb317sHCYgKk0bAG3kwO3mi
93/LXWT1cdiYVpmBcHDBAkEAmpgkFj+xZu5gWASY5ujv+FCMP0WwaH5hTnXu+tKe
PJ3d2IJZKxGnl6itKRN7GeRh9PSK0kZSqGFeVrvsJ4Nopg==
-----END RSA PRIVATE KEY-----
```

```
-----BEGIN CERTIFICATE-----
MIIC1jCCAj+gAwIBAgIBADANBgkqhkiG9w0BAQUFADBXMQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCTVMxDzANBgNVBAcMBkJpbG94aTENMAsGA1UECgwESUVURjEbMBkG
```

A1UEAwwSYmlsb3hpLmV4YW1wbGUuY29tMB4XDTA1MTAyNDA2NDAyNloXDTA2MTAy
NDA2NDAyNlowVzELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAk1TMQ8wDQYDVQQHDAZC
aWxveGkxDTALBgNVBAoMBElFVEYxGzAZBgNVBAMMEmJpbG94aS5leGFtcGxlLmNv
bTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAv6GwWC0TD47JKwKljohjwFMS
N/5trYu4seHGqgkwLhV+XC3oEO55qrizh3yIcVSKk9n9OlANJTqNjVaHyvRuzE5F
W1TYsgJS2FAxI+11iwyAn92Ry+ha//ghe8K0Pa+JL+14iSsaHVQgnfw4Qk/RqflQ
6HvB8pqjlvyrJ3q4siMCAwEAAaOBsTCBrjAdBgNVHQ4EFgQU0Z+RL47W/APDtc5B
fSoQXuEFE/wwfwYDVR0jBHgwdoAU0Z+RL47W/APDtc5BfSoQXuEFE/yhW6RZMFcx
CzAJBgNVBAYTAlVTMQswCQYDVQQIDAJNUzEPMA0GA1UEBwwGQmlsb3hpMQ0wCwYD
VQQKDARJRVRGMRswGQYDVQQDDBJiaWxveGkuZXhhbXBsZS5jb22CAQAwDAYDVR0T
BAUwAwEB/zANBgkqhkiG9w0BAQUFAAOBgQBiyKHIt8TXfGNfpnJXi5jCizOxmY8Y
gln8tyPFaeyq95TGcvTCWzdoBLVpBD+fpRWrX/II5sE6VHbbAPjjVmKbZwzQAtpp
P2Fauj28t94ZeDHN2vqzjfnHjCO24kG3Juf2T80ilp9YHcDwxjUFrt86UnlC+yid
yaTeusW5Gu7v1g==
-----END CERTIFICATE-----

Bob (bob@biloxi.example.org) now wants to send a BYE request to Alice at the end of the dialog initiated in the previous example. He therefore creates the following BYE request, which he forwards to the 'biloxi.example.org' proxy server that instantiates the authentication service role:

```
BYE sip:alice@pc33.atlanta.example.com SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnashds10
Max-Forwards: 70
From: Bob <sip:bob@biloxi.example.org>;tag=a6c85cf
To: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 231 BYE
Content-Length: 0
```

When the authentication service receives the BYE, it authenticates Bob by sending a 407 response. As a result, Bob adds an Authorization header to his request, and resends to the biloxi.example.org authentication service. Now that the service is sure of Bob's identity, it prepares to calculate an Identity header for the request. Note that this request does not have a Date header field. Accordingly, the biloxi.example.org will add a Date header to the request before calculating the identity signature. If the Content-Length header were not present, the authentication service would add it as well. The baseline message is thus:

```
BYE sip:alice@pc33.atlanta.example.com SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnashds10
Max-Forwards: 70
From: Bob <sip:bob@biloxi.example.org>;tag=a6c85cf
```

```
To: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Date: Thu, 21 Feb 2002 14:19:51 GMT
Call-ID: a84b4c76e66710
CSeq: 231 BYE
Content-Length: 0
```

[TBD: Fix example.] Also note that this request contains no Contact header field. Accordingly, biloxi.example.org will place no value in the canonical string for the addr-spec of the Contact address. Also note that there is no message body, and accordingly, the signature string will terminate, in this case, with two vertical bars. The canonical string over which the identity signature will be generated is the following (note that the first line wraps because of RFC editorial conventions):

```
sip:bob@biloxi.example.org|sip:alice@atlanta.example.com|
a84b4c76e66710|231 BYE|Thu, 21 Feb 2002 14:19:51 GMT||
```

The resulting signature (sha1WithRsaEncryption) using the private RSA key given above for biloxi.example.org, with base64 encoding, is the following:

```
sv5CTo05KqpSmtHt3dcEiO/1CWTSZtnG3iV+1nmurLXV/HmtyNS7Ltrg9dlxkWzo
eU7d7OV8HweTTDobV3itTmgPwCFjaEmMyEI3d7SyN21yNDo2ER/Ovgtw0Lu5csIp
pPqOg1uXndzHbG7mR6Rl9BnUhHufVRbp51Mn3w0gfUs=
```

Accordingly, the biloxi.example.org authentication service will create an Identity header containing that base64 signature string. It will also add an HTTPS URL where its certificate is made available. With those two headers added, the message looks like the following:

```
BYE sip:alice@pc33.atlanta.example.com SIP/2.0
Via: SIP/2.0/TLS 192.0.2.4;branch=z9hG4bKnashds10
Max-Forwards: 70
From: Bob <sip:bob@biloxi.example.org>;tag=a6c85cf
To: Alice <sip:alice@atlanta.example.com>;tag=1928301774
Date: Thu, 21 Feb 2002 14:19:51 GMT
Call-ID: a84b4c76e66710
CSeq: 231 BYE
Identity:
  "sv5CTo05KqpSmtHt3dcEiO/1CWTSZtnG3iV+1nmurLXV/HmtyNS7Ltrg9dlxkWzo
   eU7d7OV8HweTTDobV3itTmgPwCFjaEmMyEI3d7SyN21yNDo2ER/Ovgtw0Lu5csIp
   pPqOg1uXndzHbG7mR6Rl9BnUhHufVRbp51Mn3w0gfUs="
Identity-Info: <https://biloxi.example.org/biloxi.cer>;alg=rsa-sha1
Content-Length: 0
```

biloxi.example.org then forwards the request normally.

12. Privacy Considerations

The identity mechanism presented in this document is compatible with the standard SIP practices for privacy described in RFC 3323 [RFC3323]. A SIP proxy server can act both as a privacy service and as an authentication service. Since a user agent can provide any From header field value that the authentication service is willing to authorize, there is no reason why private SIP URIs that contain legitimate domains (e.g., sip:anonymous@example.com) cannot be signed by an authentication service. The construction of the Identity header is the same for private URIs as it is for any other sort of URIs.

Note, however, that for using anonymous SIP URIs, an authentication service must possess a certificate corresponding to the host portion of the addr-spec of the From header field of the request; accordingly, using domains like 'anonymous.invalid' will not be possible for privacy services that also act as authentication services. The assurance offered by the usage of anonymous URIs with a valid domain portion is "this is a known user in my domain that I have authenticated, but I am keeping its identity private". The use of the domain 'anonymous.invalid' entails that no corresponding authority for the domain can exist, and as a consequence, authentication service functions are meaningless.

RFC 3325 [RFC3325] defines the "id" priv-value token, which is specific to the P-Asserted-Identity header. The sort of assertion provided by the P-Asserted-Identity header is very different from the Identity header presented in this document. It contains additional information about the sender of a message that may go beyond what appears in the From header field; P-Asserted-Identity holds a definitive identity for the sender that is somehow known to a closed network of intermediaries that presumably the network will use this identity for billing or security purposes. The danger of this network-specific information leaking outside of the closed network motivated the "id" priv-value token. The "id" priv-value token has no implications for the Identity header, and privacy services MUST NOT remove the Identity header when a priv-value of "id" appears in a Privacy header.

Finally, note that unlike RFC 3325 [RFC3325], the mechanism described in this specification adds no information to SIP requests that has privacy implications.

13. Security Considerations

13.1. Handling of digest-string Elements

This document describes a mechanism that provides a signature over the Date header field, and either the whole or part of the To and From header fields of SIP requests, as well as optional protections for the message body. While a signature over the From header field would be sufficient to secure a URI alone, the additional headers provide replay protection and reference integrity necessary to make sure that the Identity header will not be used in cut-and-paste attacks. In general, the considerations related to the security of these headers are the same as those given in RFC 3261 [RFC3261] for including headers in tunneled 'message/sip' MIME bodies (see Section 23 in particular). The following section details the individual security properties obtained by including each of these header fields within the signature; collectively, this set of header fields provides the necessary properties to prevent impersonation.

The From header field indicates the identity of the sender of the message, and the SIP address-of-record URI, or an embedded telephone number, in the From header field is the identity of a SIP user, for the purposes of this document. The To header field provides the identity of the SIP user that this request targets. Providing the To header field in the Identity signature serves two purposes: first, it prevents cut-and-paste attacks in which an Identity header from legitimate request for one user is cut-and-pasted into a request for a different user; second, it preserves the starting URI scheme of the request, which helps prevent downgrade attacks against the use of SIPS.

The Date header field provides replay protection, as described in RFC 3261 [RFC3261], Section 23.4.2. Implementations of this specification MUST NOT deem valid a request with an outdated Date header field (the RECOMMENDED interval is that the Date header must indicate a time within 3600 seconds of the receipt of a message). The result of this is that if an Identity header is replayed within the Date interval, verifiers will recognize that it is invalid; if an Identity header is replayed after the Date interval, verifiers will recognize that it is invalid because the Date is stale.

Without the method an INVITE request could be cut- and-pasted by an attacker and transformed into a MESSAGE request without changing any fields covered by the Identity header, and moreover requests within a certain transaction could be replayed in potentially confusing or malicious ways.

RFC4474 had protections for the Contact, Call-ID and CSeq. These are removed from RFC4474bis. The absence of these header values creates some opportunities for determined attackers to impersonate based on

cut-and-paste attacks; however, the absence of these headers does not seem impactful to preventing against the simple unauthorized claiming of a From header field value.

It might seem attractive to provide a signature over some of the information present in the Via header field value(s). For example, without a signature over the sent-by field of the topmost Via header, an attacker could remove that Via header and insert its own in a cut-and-paste attack, which would cause all responses to the request to be routed to a host of the attacker's choosing. However, a signature over the topmost Via header does not prevent attacks of this nature, since the attacker could leave the topmost Via intact and merely insert a new Via header field directly after it, which would cause responses to be routed to the attacker's host "on their way" to the valid host, which has exactly the same end result. Although it is possible that an intermediary-based authentication service could guarantee that no Via hops are inserted between the sending user agent and the authentication service, it could not prevent an attacker from adding a Via hop after the authentication service, and thereby preempting responses. It is necessary for the proper operation of SIP for subsequent intermediaries to be capable of inserting such Via header fields, and thus it cannot be prevented. As such, though it is desirable, securing Via is not possible through the sort of identity mechanism described in this document; the best known practice for securing Via is the use of SIPS.

This mechanism also provides an optional signature over the bodies of SIP requests. This can help to protect non-INVITE transactions such as MESSAGE or NOTIFY, as well as INVITEs in those environments where intermediaries do not change SDP. While this is not strictly necessary to prevent the impersonation attacks, there is little purpose in establishing the identity of the user that originated a SIP request if this assurance is not coupled with a comparable assurance over the contents of the message. There are furthermore some baiting attacks (where the attacker receives a request from the target and reoriginates it to a third party) that might not be prevented by only a signature over the From, To and Date, but could be prevented by securing SDP. Note, however, that this is not perfect end-to-end security. The authentication service itself, when instantiated at an intermediary, could conceivably change the body (and SIP headers, for that matter) before providing a signature. Thus, while this mechanism reduces the chance that a replayer or man-in-the-middle will modify bodies, it does not eliminate it entirely. Since it is a foundational assumption of this mechanism that the users trust their local domain to vouch for their security, they must also trust the service not to violate the integrity of their message without good reason.

In the end analysis, the Identity, Identity-Reliance and Identity-Info headers cannot protect themselves. Any attacker could remove these headers from a SIP request, and modify the request arbitrarily afterwards. However, this mechanism is not intended to protect requests from men-in-the- middle who interfere with SIP messages; it is intended only to provide a way that the originators of SIP requests can prove that they are who they claim to be. At best, by stripping identity information from a request, a man-in-the-middle could make it impossible to distinguish any illegitimate messages he would like to send from those messages sent by an authorized user. However, it requires a considerably greater amount of energy to mount such an attack than it does to mount trivial impersonations by just copying someone else's From header field. This mechanism provides a way that an authorized user can provide a definitive assurance of his identity that an unauthorized user, an impersonator, cannot.

One additional respect in which the Identity-Info header cannot protect itself is the 'alg' parameter. The 'alg' parameter is not included in the digest-string, and accordingly, a man-in-the-middle might attempt to modify the 'alg' parameter. However, it is important to note that preventing men-in-the-middle is not the primary impetus for this mechanism. Moreover, changing the 'alg'

would at worst result in some sort of bid-down attack, and at best cause a failure in the verifier. Note that only one valid 'alg' parameter is defined in this document and that thus there is currently no weaker algorithm to which the mechanism can be bid down. 'alg' has been incorporated into this mechanism for forward-compatibility reasons in case the current algorithm exhibits weaknesses, and requires swift replacement, in the future.

13.2. Display-Names and Identity

As a matter of interface design, SIP user agents might render the display-name portion of the From header field of a caller as the identity of the caller; there is a significant precedent in email user interfaces for this practice. As such, it might seem that the lack of a signature over the display-name is a significant omission.

However, there are several important senses in which a signature over the display-name does not prevent impersonation. In the first place, a particular display-name, like "Jon Peterson", is not unique in the world; many users in different administrative domains might legitimately claim that name. Furthermore, enrollment practices for SIP-based services might have a difficult time discerning the legitimate display-name for a user; it is safe to assume that impersonators will be capable of creating SIP accounts with arbitrary display-names. The same situation prevails in email today. Note

that an impersonator who attempted to replay a message with an Identity header, changing only the display-name in the From header field, would be detected by the other replay protection mechanisms described in Section 13.1.

Of course, an authentication service can enforce policies about the display-name even if the display-name is not signed. The exact mechanics for creating and operationalizing such policies is outside the scope of this document. The effect of this policy would not be to prevent impersonation of a particular unique identifier like a SIP URI (since display-names are not unique identifiers), but to allow a domain to manage the claims made by its users. If such policies are enforced, users would not be free to claim any display-name of their choosing. In the absence of a signature, man-in-the-middle attackers could conceivably alter the display-names in a request with impunity. Note that the scope of this specification is impersonation attacks, however, and that a man-in-the-middle might also strip the Identity and Identity-Info headers from a message.

There are many environments in which policies regarding the display-name aren't feasible. Distributing bit-exact and internationalizable display-names to end-users as part of the enrollment or registration process would require mechanisms that are not explored in this document. In the absence of policy enforcement regarding domain names, there are conceivably attacks that an adversary could mount against SIP systems that rely too heavily on the display-name in their user interface, but this argues for intelligent interface design, not changes to the mechanisms. Relying on a non-unique identifier for identity would ultimately result in a weak mechanism.

13.3. Securing the Connection to the Authentication Service

The assurance provided by this mechanism is strongest when a user agent forms a direct connection, preferably one secured by TLS, to an intermediary-based authentication service. The reasons for this are twofold:

If a user does not receive a certificate from the authentication service over this TLS connection that corresponds to the expected domain (especially when the user receives a challenge via a mechanism such as Digest), then it is possible that a rogue server is attempting to pose as an authentication service for a domain that it does not control, possibly in an attempt to collect shared secrets for that domain. A similar practice could be used for telephone numbers, though the application of certificates for telephone numbers to TLS is left as a matter for future study.

Without TLS, the various header field values and the body of the request will not have integrity protection when the request arrives at an authentication service. Accordingly, a prior legitimate or illegitimate intermediary could modify the message arbitrarily.

Of these two concerns, the first is most material to the intended scope of this mechanism. This mechanism is intended to prevent impersonation attacks, not man-in-the-middle attacks; integrity over the header and bodies is provided by this mechanism only to prevent replay attacks. However, it is possible that applications relying on the presence of the Identity header could leverage this integrity protection, especially body integrity, for services other than replay protection.

Accordingly, direct TLS connections SHOULD be used between the UAC and the authentication service whenever possible. The opportunistic nature of this mechanism, however, makes it very difficult to constrain UAC behavior, and moreover there will be some deployment architectures where a direct connection is simply infeasible and the UAC cannot act as an authentication service itself. Accordingly, when a direct connection and TLS are not possible, a UAC should use the SIPS mechanism, Digest 'auth-int' for body integrity, or both when it can. The ultimate decision to add an Identity header to a request lies with the authentication service, of course; domain policy must identify those cases where the UAC's security association with the authentication service is too weak.

13.4. Domain Names, Certificates and Subordination

When a verifier processes a request containing an Identity-Info header with a domain signature, it must compare the domain portion of the URI in the From header field of the request with the domain name that is the subject of the certificate acquired from the Identity-Info header. While it might seem that this should be a straightforward process, it is complicated by two deployment realities. In the first place, certificates have varying ways of describing their subjects, and may indeed have multiple subjects, especially in 'virtual hosting' cases where multiple domains are managed by a single application. Secondly, some SIP services may delegate SIP functions to a subordinate domain and utilize the procedures in RFC 3263 [RFC3263] that allow requests for, say, 'example.com' to be routed to 'sip.example.com'. As a result, a user with the AoR 'sip:jon@example.com' may process requests through a host like 'sip.example.com', and it may be that latter host that acts as an authentication service.

To meet the second of these problems, a domain that deploys an authentication service on a subordinate host MUST be willing to supply that host with the private keying material associated with a certificate whose subject is a domain name that corresponds to the domain portion of the AoRs that the domain distributes to users. Note that this corresponds to the comparable case of routing inbound SIP requests to a domain. When the NAPTR and SRV procedures of RFC 3263 are used to direct requests to a domain name other than the domain in the original Request-URI (e.g., for 'sip:jon@example.com', the corresponding SRV records point to the service 'sip1.example.org'), the client expects that the certificate passed back in any TLS exchange with that host will correspond exactly with the domain of the original Request-URI, not the domain name of the host. Consequently, in order to make inbound routing to such SIP services work, a domain administrator must similarly be willing to share the domain's private key with the service. This design decision was made to compensate for the insecurity of the DNS, and it makes certain potential approaches to DNS-based 'virtual hosting' unsecurable for SIP in environments where domain administrators are unwilling to share keys with hosting services.

A verifier MUST evaluate the correspondence between the user's identity and the signing certificate by following the procedures defined in RFC 2818 [RFC2818], Section 3.1. While RFC 2818 [RFC2818] deals with the use of HTTP in TLS, the procedures described are applicable to verifying identity if one substitutes the "hostname of the server" in HTTP for the domain portion of the user's identity in the From header field of a SIP request with an Identity header.

Because the domain certificates that can be used by authentication services need to assert only the hostname of the authentication service, existing certificate authorities can provide adequate certificates for this mechanism. However, not all proxy servers and user agents will be able to support the root certificates of all certificate authorities, and moreover there are some significant differences in the policies by which certificate authorities issue their certificates. This document makes no recommendations for the usage of particular certificate authorities, nor does it describe any particular policies that certificate authorities should follow, but it is anticipated that operational experience will create de facto standards for authentication services. Some federations of service providers, for example, might only trust certificates that have been provided by a certificate authority operated by the federation. It is strongly RECOMMENDED that self-signed domain certificates should not be trusted by verifiers, unless some previous key exchange has justified such trust.

[TBD: DANE?]

For further information on certificate security and practices, see RFC 3280 [RFC3280]. The Security Considerations of RFC 3280 [RFC3280] are applicable to this document.

13.5. Authorization and Transitional Strategies

Ultimately, the worth of an assurance provided by an Identity header is limited by the security practices of the domain that issues the assurance. Relying on an Identity header generated by a remote administrative domain assumes that the issuing domain used its administrative practices to authenticate its users. However, it is possible that some domains will implement policies that effectively make users unaccountable (e.g., ones that accept unauthenticated registrations from arbitrary users). The value of an Identity header from such domains is questionable. While there is no magic way for a verifier to distinguish "good" from "bad" domains by inspecting a SIP request, it is expected that further work in authorization practices could be built on top of this identity solution; without such an identity solution, many promising approaches to authorization policy are impossible. That much said, it is RECOMMENDED that authentication services based on proxy servers employ strong authentication practices such as token-based identifiers.

One cannot expect the Identity and Identity-Info headers to be supported by every SIP entity overnight. This leaves the verifier in a compromising position; when it receives a request from a given SIP user, how can it know whether or not the sender's domain supports Identity? In the absence of ubiquitous support for identity, some transitional strategies are necessary.

A verifier could remember when it receives a request from a domain that uses Identity, and in the future, view messages received from that domain without Identity headers with skepticism.

A verifier could query the domain through some sort of callback system to determine whether or not it is running an authentication service. There are a number of potential ways in which this could be implemented; use of the SIP OPTIONS method is one possibility. This is left as a subject for future work.

In the long term, some sort of identity mechanism, either the one documented in this specification or a successor, must become mandatory-to-use for the SIP protocol; that is the only way to guarantee that this protection can always be expected by verifiers.

Finally, it is worth noting that the presence or absence of the Identity headers cannot be the sole factor in making an authorization decision. Permissions might be granted to a message on the basis of

the specific verified Identity or really on any other aspect of a SIP request. Authorization policies are outside the scope of this specification, but this specification advises any future authorization work not to assume that messages with valid Identity headers are always good.

14. IANA Considerations

[TBD: update for rfc4474bis or remove?]

This document requests changes to the header and response-code sub-registries of the SIP parameters IANA registry, and requests the creation of two new registries for parameters for the Identity-Info header.

14.1. Header Field Names

This document specifies two new SIP headers: Identity and Identity-Info. Their syntax is given in Section 10. These headers are defined by the following information, which has been added to the header sub-registry under http://www.iana.org/assignments/sip-parameters

Header Name: Identity
Compact Form: y
Header Name: Identity-Info
Compact Form: n

14.2. 428 'Use Identity Header' Response Code

This document registers a new SIP response code, which is described in Section 5.2. It is sent when a verifier receives a SIP request that lacks an Identity header in order to indicate that the request should be re-sent with an Identity header. This response code is defined by the following information, which has been added to the method and response-code sub-registry under http://www.iana.org/assignments/sip-parameters

Response Code Number: 428
Default Reason Phrase: Use Identity Header

14.3. 436 'Bad Identity-Info' Response Code

This document registers a new SIP response code, which is described in Section 5.2. It is used when the Identity-Info header contains a URI that cannot be dereferenced by the verifier (either the URI scheme is unsupported by the verifier, or the resource designated by the URI is otherwise unavailable). This response code is defined by the following information, which has been added to the method and response-code sub-registry under http://www.iana.org/assignments/sip-parameters

Response Code Number: 436
Default Reason Phrase: Bad Identity-Info

14.4. 437 'Unsupported Certificate' Response Code

This document registers a new SIP response code, which is described in Section 5.2. It is used when the verifier cannot validate the certificate referenced by the URI of the Identity-Info header, because, for example, the certificate is self-signed, or signed by a root certificate authority for whom the verifier does not possess a root certificate. This response code is defined by the following information, which has been added to the method and response-code sub-registry under http://www.iana.org/assignments/sip-parameters

Response Code Number: 437
Default Reason Phrase: Unsupported Certificate

14.5. 438 'Invalid Identity Header' Response Code

This document registers a new SIP response code, which is described in Section 5.2. It is used when the verifier receives a message with an Identity signature that does not correspond to the digest-string calculated by the verifier. This response code is defined by the following information, which has been added to the method and response-code sub-registry under http://www.iana.org/assignments/sip-parameters

Response Code Number: 438
Default Reason Phrase: Invalid Identity Header

14.6. Identity-Info Parameters

The IANA has created a new registry for Identity-Info headers. This registry is to be prepopulated with a single entry for a parameter

called 'alg', which describes the algorithm used to create the signature that appears in the Identity header. Registry entries must contain the name of the parameter and the specification in which the parameter is defined. New parameters for the Identity-Info header may be defined only in Standards Track RFCs.

14.7. Identity-Info Algorithm Parameter Values

The IANA has created a new registry for Identity-Info 'alg' parameter values. This registry is to be prepopulated with a single entry for a value called 'rsa-sha1', which describes the algorithm used to create the signature that appears in the Identity header. Registry entries must contain the name of the 'alg' parameter value and the specification in which the value is described. New values for the 'alg' parameter may be defined only in Standards Track RFCs.

15. Acknowledgements

The authors would like to thank the many commentators on this document.

16. Original RFC 4474 Requirements

The following requirements were crafted throughout the development of the mechanism described in this document. They are preserved here for historical reasons.

> The mechanism must allow a UAC or a proxy server to provide a strong cryptographic identity assurance in a request that can be verified by a proxy server or UAS.
>
> User agents that receive identity assurances must be able to validate these assurances without performing any network lookup.
>
> User agents that hold certificates on behalf of their user must be capable of adding this identity assurance to requests.
>
> Proxy servers that hold certificates on behalf of their domain must be capable of adding this identity assurance to requests; a UAC is not required to support this mechanism in order for an identity assurance to be added to a request in this fashion.
>
> The mechanism must prevent replay of the identity assurance by an attacker.

In order to provide full replay protection, the mechanism must be capable of protecting the integrity of SIP message bodies (to ensure that media offers and answers are linked to the signaling identity).

It must be possible for a user to have multiple AoRs (i.e., accounts or aliases) that it is authorized to use within a domain, and for the UAC to assert one identity while authenticating itself as another, related, identity, as permitted by the local policy of the domain.

17. Changes from RFC4474

17.1. Motivation for Changes

The original sip-identity drafts that lead to RFC 4474 [RFC4474] were first published in 2002. Since that point many things have changed that impact the design.

- The DNS root has been signed.
- SPAM continues to be a problem.
- It has become clear that B2BUAs will continue to be a major factor in SIP deployments.
- Multipart MIME has failed as a SIP extension mechanism.
- Widespread identity providers such as Facebook have emerged.
- Techniques for non-carrier entities to verify phone numbers and then use them for addressing (such as Apple's iMessage) have been shown to be commercially feasible.
- Substantial portions of commercial, government, and personal voice communications rely on SIP at some stage in the communications.
- The cost of operating large databases has fallen and outsourced versions of these databases have become cheaply available.
- Extensive experience and user research has improved our understanding of how to present security information to users.
- The world is in the middle of a huge transition to mobile devices. Even the most limited modern mobile devices have user interface and computational capabilities that greatly exceed a 2002-era SIP phone.

The authors believe that the confluence of changing technology, the evolution of mobile devices and internet, and a political will to change make this the right time to consider an change of the scope of 4474 to solve the following problems:

- Assert strong identity for E.164 numbers such as +1 408 555-1212
- Continue to assert strong identity for domain scoped names such as alice@example.com
- Work for calls crossing even the most adverse networks such as the PSTN.
- Provide reliable information about who is calling before the call is answered to help stop SPAM.
- Provide reliable information about who you are talking to.
- Work with evolving non SIP based communications systems such as WebRTC.
- Potentially, as future work explore organization attributes (e.g., "this is a Bank").

We believe it is possible to solve all of these in a way that is commercially viable, deployable, and provides a delightful user experience.

The core problem in a global identity system with delegated names is understanding who is authorized to make assertions about a given name. The proposal is to solve that problem with a two pronged approach. The design of such a system is outside the scope of this draft, and perhaps of the IETF, but we believe it will have a twofold character:

First, it will delegate responsibility for a number down from a root in a series of delegation sub delegation towards the user. For example, the North American Numbering Plan Administrator assigns a portion of the +1 space to a service provider. That service provider may assign a sub space to a company and that company may assign a number to a user. At each level of delegation, cryptographic credentials could be provided that allow the user to prove the space was delegated to them given some common trust root. This approach is referred to as "delegation" and effectively works from the top down.

The other prong to solving the problem is called "claims" and works via a bottom up approach. The end user of a number basically claims it and some trusted system validates this claim. The validation may

be as simple as sending a SMS to the number or more complicated such as the VIPR system.

The delegation approach creates an easier user experience but is harder to deploy from a business incentive point of view so our approach is to do both and work down from the top and up from the bottom with a meet in the middle approach to coverage of the full name space. For the purposes of the current work, it is envisioned that a certificate authority could encompass both approaches.

Authentication services that possess a credential (whether of the delegation or claim variety) for a telephone number or domain name can, in this mechanism, create one of two types of assertions: basic assertions and reliance assertions. The basic assertion provides replay protection, whereas the reliance assertion provides a broader body protection. Some networks might modify the signaling in ways that impact the reliance assertions but not the other, and thus the reliance assertion is optional.

As in RFC4474, identity assertions are passed in-band in SIP from the caller to the callee for verification. There are however some cases where in-band signaling cannot survive the call path, such as when the call passes through a gateway to the PSTN. This specification assumes that other, out-of-band mechanisms may be used in cases where in-band identity is not carried end-to-end, but those mechanisms are outside the scope of this document.

17.2. Changes to the Identity-Info Header

RFC4474 restricted the subject of the certificate to a domain name, and accordingly the RFC4474 Identity-Info header contains a URI which designates a certificate whose subject (more precisely, subjectAltName) must correspond to the domain of the URI in the From header field value of the SIP request. Per the analysis in [I-D.peterson-secure-origin-ps], this document relaxes that constraint to allow designating an alternative authority for telephone numbers, when telephone numbers appear in the From header field value.

These changes will allow the Identity-Info URI to point to the certificate with authority over the calling telephone number. A verification service will therefore authorize a SIP request when the telephone number in the From header field value agrees with the subject of the certificate. Verification services must of course trust the certificate authority that issued the certificate in question. To implement this change to the Identity-Info header, we must allow for two possibilities for the conveyance of a telephone number in a request: appearing within a tel URI or appearing as the

user portion of a SIP URI. Therefore, we must prescribe the verification service behind in the case where the From header field value URI contains a telephone user part followed by a domain -- which should the verification service expect to find in a certificate?

Future version of this document may explore alternate ways of acquiring credentials, including the use of credentials other than certificates. This might include implementing enough flexibility in the URI to allow a model more like the IdP model described in [I-D.rescorla-rtcweb-generic-idp]; this could be useful as RTCWeb sees increasing deployment. We also should consider any implications of the signing of the DNSSEC root and the DANE specifications to the existing Identity-Info uses with domain name. At a high level, it is not expected that the proposed changes will radically alter the semantics of Identity-Info.

17.3. Changes to the Identity Header

Per the analysis in [I-D.peterson-secure-origin-ps], this document changes the signature mechanism that RFC44474 specified for the Identity header: in particular, to replace this signature mechanism with one that is more likely to survive end-to-end in SIP networks where intermediaries act as back-to-back user agents rather than proxy servers.

To accomplish this, we here create two distinct signatures within SIP requests: a basic assurance and a reliance assurance. The basic assurance prevents impersonation attacks by providing a signature over the From header field value and certain other headers which will allow a verification service to detect some cut-and-paste attacks. The reliance assurance protects against attackers changing other parameters of the call: these include the entirely of the messaging body, including the target IP address and ports in SDP which, if unprotected, can allow an attacker to succeed with more sophisticated cut-and-paste attacks. Authentication services behavior would change to allow them to decide, based on their policy in a deployment environment, whether only the basic assurance can realistically survive network transit, or if the reliance assurance should be available. There are several similar design choices in this space to consider, and some analysis will be required to identify the best option.

In cases where the From header field value of a SIP request contains a SIP URI with a telephone number user part, we will also consider replay assurance canonicalizations that do not cover the domain portion of the URI.

[TBD: in order to preserve critical security parameters even in adverse network conditions, should the basic assurance integrity protection must always cover security parameters of the SDP required to negotiate media-level security? There may be other exception cases, or extensibility mechanisms, worth considering here.]

18. References

18.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3280] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

[RFC3370] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, August 2002.

[RFC3548] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003.

[RFC3893] Peterson, J., "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format", RFC 3893, September 2004.

[RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

18.2. Informative References

[I-D.cooper-iab-secure-origin-00] Cooper, A., Tschofenig, H., Peterson, J., and B. Aboba, "Secure Call Origin Identification", draft-cooper-iab-secure-origin-00 (work in progress), November 2012.

[I-D.peterson-secure-origin-ps]
Peterson, J., Schulzrinne, H., and H. Tschofenig, "Secure Origin Identification: Problem Statement, Requirements, and Roadmap", draft-peterson-secure-origin-ps-00 (work in progress), May 2013.

[I-D.peterson-sipping-retarget]
Peterson, J., "Retargeting and Security in SIP: A Framework and Requirements", draft-peterson-sipping-retarget-00 (work in progress), February 2005.

[I-D.rescorla-callerid-fallback]
Rescorla, E., "Secure Caller-ID Fallback Mode", draft-rescorla-callerid-fallback-00 (work in progress), May 2013.

[I-D.rescorla-rtcweb-generic-idp]
Rescorla, E., "RTCWEB Generic Identity Provider Interface", draft-rescorla-rtcweb-generic-idp-01 (work in progress), March 2012.

[RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[RFC2585] Housley, R. and P. Hoffman, "Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP", RFC 2585, May 1999.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

[RFC3761] Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 3761, April 2004.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC4475] Sparks, R., Hawrylyshen, A., Johnston, A., Rosenberg, J., and H. Schulzrinne, "Session Initiation Protocol (SIP) Torture Test Messages", RFC 4475, May 2006.

[RFC6919] Barnes, R., Kent, S., and E. Rescorla, "Further Key Words for Use in RFCs to Indicate Requirement Levels", RFC 6919, April 1 2013.

Authors' Addresses

Jon Peterson
NeuStar

Email: jon.peterson@neustar.biz

Cullen Jennings
Cisco
400 3rd Avenue SW, Suite 350
Calgary, AB T2P 4H2
Canada

Email: fluffy@iii.ca

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, CA 94303
USA

Phone: +1 650 678 2350
Email: ekr@rtfm.com